Distinguishing Data Transience from False Injection in Sensor Networks

Vinod Shukla and Daji Qiao
Iowa State University, Ames, IA 50011
{vkshukla, daji}@iastate.edu

***Abstract*— Wireless sensor networks are increasingly being employed for potentially hazardous and critical applications such as monitoring the gas concentration levels in a battle field. In such sensitive applications, it is vital to monitor closely the transient phenomenon and take requisite preventive and corrective actions, if necessary. In such scenarios, due to the presence of adversaries who intend to disrupt the functioning of the system, it becomes imperative to shield our system from false data injection attacks. We propose a novel secure statistical scheme, called SSTF to effectively monitor the transient phenomenon while being immune to false data injection attacks. For achieving our goals, we require the sensors to do a lightweight computation and report a simple statistical digest in addition to the current sensed reading. SSTF is a two-tier system consisting of a statistical inter-sensor testing framework, which is the kernel of our scheme, employed atop an enhanced version of a well-known existing security scheme. We present detailed theoretical analysis and in-depth simulations to show the effectiveness of SSTF.**

I. Introduction

Sensor nodes may be deployed in hostile environments and due to the sheer magnitude of number of nodes deployed in a network, it is infeasible to physically monitor all of them. As such, the network and sensor nodes are susceptible to various kinds of attacks from adversaries. Particularly, the nodes may be captured or compromised, and all the secret information stored in the nodes would be known to the adversary, who can then easily inject false reports about the phenomenon to be sensed into the network. Such an attack is called false data injection attack [1].

The issue of preventing false data injection attack has attracted substantial research interests [1]–[5]. Most existing schemes assume that each individual sensor reports only the sensed reading. So, if the values reported by sensors do not agree to each other, the data is considered false and rejected by some process akin to majority voting where all other sensors should agree. Consider a scenario where the phenomenon to be sensed has transient temporal and spatial variations. In this case, different sensors may sense different readings and may not agree to each other all the time. Such transient data though genuine and important, will be classified by existing schemes as false and rejected. Motivated by this observation, we address the *distinction between genuine transient data vis-a-vis false data* in this paper.

Sensor networks are typically organized into clusters. Each cluster has a Cluster Head (CH) responsible for collecting data from sensors in the cluster, doing aggregation and forwarding to a distant Base Station (BS). We propose SSTF, a novel **S**ecure **S**tatistical scheme to distinguish data **T**ransience from **F**alse injection in a clustered wireless sensor network. The key ideas of SSTF are twofold. Firstly, each sensor computes a statistical digest of the monitored phenomenon over a moving window of recent readings and reports this digest along with the current reading to CH. By utilizing the statistical digests to aid in decision making and data aggregation at the CH, SSTF is able to distinguish transient data from false data in most scenarios, which is very difficult if only the current sensed readings are reported by individual sensors. Secondly, SSTF requires the CH to perform a series of carefully-designed inter-sensor tests on both readings and digests reported by individual sensors. Since, the false data reported by the compromised node has to pass the inter-sensor tests to escape detection, the impact of false data on the network is significantly restricted. The security framework of SSTF is based on IHHAS [1].

The rest of the paper is organized as follows. We discuss the related work in Section II and give the system model and problem statement in Section III. We describe the proposed SSTF scheme in Section IV, present a realization of SSTF using IHHAS in Section V and analyze its security performance in Section VI. Simulation results are presented in Section VII. Finally, we present conclusions and future work in Section VIII.

II. Related Work

In this section, we present some relevant research pertaining to false data rejection and secure aggregation.

1) False Data Rejection: Ye et al [2] propose a statistical en-route filtering scheme (SEF), which allows both

1-4244-1268-4/07/$25.00 ©2007 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE SECON 2007 proceedings.

BS and en-route nodes to detect false data with a certain probability. Zhu et al [1] propose an **I**nterleaved **H**op-by-**H**op **A**uthentication **S**cheme (IHHAS) where pairwise keys are established between nodes $t+1$ hops away and up to t compromised nodes can be tolerated. Yang et al [3] present a commutative cipher based en-route filtering scheme (CCEF) which is based on public-key algorithms that have been reported not suitable for sensor networks due to limited resource capacity of sensor nodes [6]. Yu et al [4] present a dynamic en-route filtering scheme for filtering false data injection; alleviating the constraint of fixed path requirement between BS and CH in [1], [3]; thus, making the scheme better suited to deal with dynamic topology of sensor networks. Zhang et al [7] present the interleaved authentication for filtering false data in multipath routing based sensor networks.

2) Secure Aggregation: Przydatek et al [5] present SIA for secure aggregation in sensor networks. It focuses on reducing trust in BS, when queried by a trusted outside user and gives schemes to compute a few aggregation primitives. Mahimkar et al [8] present SecureDAV which uses Merkle Hash Trees to avoid over-reliance on CH. Since attacker does not know the cluster key, it cannot generate the full signature. Assuming a trusted BS, Wagner [9] discusses which aggregation functions can be meaningfully computed with resilience. However, [9] does not consider any in-network aggregation and only BS does the aggregation. Hu and Evans [10] propose a secure hop-by-hop aggregation scheme that works if one node is compromised. Yang et al [11] present SDAP which partitions sensor nodes in a tree topology and performs a commitment-based hop-by-hop aggregation in each subtree to generate a group aggregated result.

3) Novelty of our work: There is significant difference between past research and our work. In general, past research has focused on computing the aggregates (a single value e.g. sum, count, min, max, etc) securely or accepting the aggregate being correct to a certain probability. Similarly, false data rejection protocols involve accepting or rejecting single values which are proven equal (with some tolerance) or not equal to each other. As discussed earlier, this leads to rejection of even genuine but transient data and we cannot observe the variations in the phenomenon being sensed. Thus, we focus on solving a novel problem: how to observe a time-variant phenomenon by accepting the genuine transient data and at the same time limit the effect of false data. In general, it is difficult to distinguish between transient and false data if sensor reading is the only information reported. In our scheme, the sensor nodes also send a simple statistical digest along with the reading to CH as opposed to sending only the reading in the existing schemes.

III. Models and Problem Statement

A. System Model

We consider a clustered wireless sensor network that is partitioned into distinct clusters after deployment. Each cluster has a Cluster Head (CH) and a set of sensor nodes, which gather information and transmit it to CH. CH does decision making and aggregation on the information received from the sensors and forwards an aggregated report to a distant Base Station (BS). We do not address clustering-related issues such as CH selection or rotation in this work.

Distinct clusters could be sensing different phenomena, however we assume that all sensors in a single cluster sense the same phenomenon. The sampling rate of sensors is dependent on the maximum temporal change in the phenomenon as well as the maximum spatial diffusion rate. Instead of sending only the sensed reading to CH, each sensor does a lightweight computation over a moving window of recent sensed readings and sends a simple statistical digest to CH at periodic intervals.

B. Threat Model

Sensor nodes may be compromised or physically captured. All secret information stored in compromised nodes can be accessed by adversaries and they can launch multiple attacks like dropping or altering the message contents going through them, so as to prevent BS from receiving authentic sensor readings. Also, there may be colluded attacks where two or more nodes collaborate to let the false reports escape detection en-route to BS.

C. Problem Statement

Due to time and space variant nature of the phenomenon being sensed, instantaneous sensor readings recorded by individual sensors in a cluster may vary. In a monitoring application, it is often critical to observe such transient but genuine data and report them with low false positives. On the other hand, a compromised node (or group of colluding compromised nodes) will try to inject a false reading into the network and our aim is to minimize the impact of false injection on the aggregation process and detect it eventually. Thus, we identify the following design goals for our scheme: (i) It should distinguish genuine transient data from injected false data and report them with low false positives; (ii) False data injection should have minimal impact on the aggregation process and be detected as soon as possible; and (iii) It should tolerate a large number of compromised nodes.

IV. Proposed SSTF Scheme

SSTF is in short for **S**ecure **S**tatistical scheme to distinguish data **T**ransience from **F**alse injection. Our proposed scheme is a two-tier system with a statistical framework on top of a security framework. Such modular design enables us to integrate the statistical framework on top of any existing security scheme with necessary adaptation and enhancement. Section IV-A presents our proposed statistical framework and Section V describes one particular realization of the SSTF scheme by integrating the proposed framework with an enhanced version of a well-known security scheme, IHHAS [1].

A. Statistical Framework

Statistical framework is the kernel of our proposed scheme. This framework can be applied on top of any security scheme to achieve the goal of preserving data transience while being immune to false data injection attacks. The statistical framework can be divided into four categories of operation: *Individual Sensor Behavior*; *Cluster Head Behavior*; *Sensor Endorsement*; and *Enroute Nodes and Base Station Behavior*. Table I summarizes the notations used in this section.

TABLE I
Notations for Statistical Framework

Notation	Remarks
$\mathcal{P}$	Phenomenon being sensed by a cluster
$\mathcal{D}$	Minimum diffusion rate of $\mathcal{P}$, measured in units/sec.
ρ	Phenomenon variation rate: maximum change in the phenomenon per unit time measured in units/sec (e.g. ppm/sec for gas concentration, etc).
x	Sampling rate at each sensor, measured in samples/sec.
d	Maximum distance between any two sensor nodes within a cluster, measured in meters.
n	Reporting interval: each sensor sends report to CH every n samples.
SW_i	Sliding window for the i-th report.
w	Size of the sliding window.
τ	Number of nodes in the cluster (including CH)
v_k	A sensor in the cluster.
r_{ki}	Sensed reading reported by v_k in the i-th report.
μ_{ki}	Sample mean reported by v_k in the i-th report.
σ^2_{ki}	Sample variance reported by v_k in the i-th report.
R_{ki}	The i-th report sent by v_k in the format of $\langle r_{ki}, \mu_{ki}, \sigma^2_{ki}\rangle$.
R_{Ag_i}	The i-th aggregated report generated by CH in the format of $\langle r_{Ag_i}, \mu_{Ag_i}, \sigma^2_{Ag_i}\rangle$.

1) Individual Sensor Behavior: A sensor node senses the phenomenon at the sampling rate. It maintains a buffer size equal to that of the sliding window (w) to store the w most recent readings. Every time a new reading is sensed, the oldest one is deleted; thus a sliding window of size w is implemented at each sensor. We need to have w samples to generate a report. After every reporting interval (n samples), the sensor node v_k computes a simple statistical digest consisting of the sample mean, μ_{ki} and sample variance, σ^2_{ki} over the sliding window SW_i. This is further illustrated in Fig. 1. The report from sensor node v_k to CH is in the format of $R_{ki} \equiv \langle r_{ki}, \mu_{ki}, \sigma^2_{ki}\rangle$.

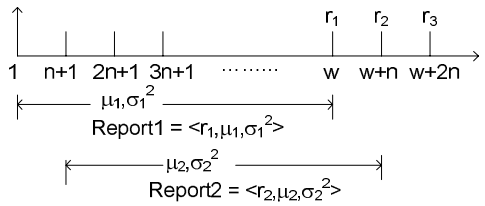


Fig. 1. Sliding window implementation and report generation at a particular sensor node. Hence, the sensor index k has been omitted. r_i, μ_i, σ_i are respectively the last reading, mean and standard deviation of w samples in SW_i. Shown are reports for two windows (i = 1, 2). There are n non-overlapping samples between two adjacent windows.

2) Cluster Head Behavior: In addition to performing the same functions as other sensors in the cluster, CH collects the reports R_{ki} from all individual sensors for testing and aggregation. CH performs two inter-sensor tests. First, CH does a *distribution test* to verify the conformity of the reported digests. Next, by utilizing the reported digests, CH does a *bin test* on the reports that pass the distribution test to limit the impact of false data.

Distribution Test: CH does pairwise tests to check whether the distributions $N_{ki}(\mu_{ki}, \sigma^2_{ki})$ $(1 \leqslant k \leqslant q)$ conform to each other, where $q \leqslant \tau$ is the number of reporting nodes. A minimum of p nodes need to pass distribution test for the aggregation to proceed. The number p is discussed in Section V. CH takes the means μ_{ki} reported by the sensors as measurements of a common mean. For two sensors v_j and v_k, CH does a *z-test* [12] to check whether the means μ_{ji} and μ_{ki} are the same with $\alpha\%$ confidence level, where α is a design parameter and the desired α can be achieved by adjusting the sliding window size. The z-test checks whether:

- $|\mu_{ji} - \mu_{ki}| \leqslant z_\alpha \times \frac{\sigma^2_{ji}+\sigma^2_{ki}}{\sqrt{w}}$.

If γ $(p \leqslant \gamma \leqslant q)$ sensors pass this test, CH proceeds to calculate the aggregated mean and variance based on the sample means and variances reported by the individual sensor nodes that have passed the distribution test. Specifically, CH takes the means reported by individual sensors as measurements of a common aggregated mean that needs to be computed. Under this assumption, the

aggregated mean and variance can be computed by using Maximum Likelihood Estimation (MLE):

$$\begin{cases} \mu_{Ag_i} = \frac{\sum_{k=1}^{\gamma} \mu_{ki}/\sigma_{ki}^2}{\sum_{k=1}^{\gamma} 1/\sigma_{ki}^2}, \\ \sigma_{Ag_i}^2 = (\sum_{k=1}^{\gamma} 1/\sigma_{ki}^2)^{-1}. \end{cases} \quad (1)$$

Bin Test: We utilize the aggregated variance produced at the end of distribution test to limit impact of false data on the aggregation process with the following bin test. The bin test is performed only on the readings reported by individual sensors that have passed the distribution test, called the *eligible sensors*. The intuition behind bin test is that all sensors observe the same phenomenon which is a diffusion process, so the difference between genuine readings reported by any two sensors is most likely to be less than twice the aggregated standard deviation, σ_{Ag_i}. Hence, for each eligible sensor v_k, CH utilizes σ_{Ag_i} to form a bin of size $[r_{ki} - 2\sigma_{Ag_i}, r_{ki} + 2\sigma_{Ag_i}]$. Then it checks if the reading reported by other eligible sensors lie in this bin. CH does this for every eligible sensor. Once it knows the bin size of all eligible nodes, it picks the one with largest size and averages the readings to compute a final aggregated reading r_{Ag_i}.

Finally, CH generates the i-th aggregated report $R_{Ag_i} \equiv \langle r_{Ag_i}, \mu_{Ag_i}, \sigma_{Ag_i}^2 \rangle$ and sends to the individual sensors belonging to the selected bin for endorsement.

3) Sensor Endorsement: Sensor endorsement is done to prevent CH from lying about the aggregation process. Specifically, when a sensor v_k receives the aggregated report R_{Ag_i} from CH for endorsement, it performs the following tests:

- whether $\sigma_{Ag_i} \leqslant \sigma_{ki}$;
- z-test to test whether $\mu_{Ag_i} = \mu_{ki}$;
- whether $r_{Ag_i} \in [r_{ki} - 2 \times \sigma_{Ag_i}, r_{ki} + 2 \times \sigma_{Ag_i}]$.

If the above conditions are met. v_k endorses R_{Ag_i} using two keys, one it shares with an en-route node and one with BS as described in Section V. v_k sends this endorsed report to CH. Since CH does not have knowledge of any of these two keys, it can make no further changes to the endorsed reports.

4) En-route Nodes and Base Station Behavior: When CH receives endorsements from individual sensors, it merges them into a single report and forwards it to BS. When an en-route node receives the report, it verifies the integrity of the report by checking the endorsement. If it is able to verify, it forwards the report to the next en-route node, else it drops the report. The process thus continues to the BS. If the verification at BS succeeds, the report R_{Ag} is accepted, else it is discarded. BS records all the reports from each CH in the network, and uses them to depict the variations in the phenomenon. En-route nodes and BS as such, have no major role in the statistical framework.

V. Realization of SSTF using IHHAS as the security scheme

Here we present a complete realization of SSTF using an existing security scheme IHHAS [1]. IHHAS is not directly applicable so we enhance the scheme to meet our requirements. First, we present an overview of IHHAS, then describe the limitations and finally present the complete SSTF realization with modified IHHAS.

Similar to IHHAS we make the following security assumptions. Every node shares a master secret key with BS. Each node knows its one-hop neighbors. Pairwise keys can be established between next-hop nodes or nodes that are multiple hops away. All nodes are equally trusted and if a node is compromised, all the information it holds will also be compromised. It is assumed that BS is not compromised. We consider a clustered sensor network and there can be either one-to-one or many-to-one correspondence established between the cluster nodes and the en-route nodes to BS. With a proper association scheme and en-route filtering scheme, it is ensured that as long as a valid cluster node does not sign the false aggregated report, it will eventually be detected enroute and dropped. Table II summarizes the notations that will be used in this section.

TABLE II
Notations for Security Framework

Notation	Remarks
K_u	Key shared between node u and BS.
K_{uv}	Pairwise key shared between nodes u and v.
F	Family of pseudo-random functions.
K_u^a	Node u's authentication key: $K_u^a = F_{K_u}(0)$.
u_i $(1 \leqslant i \leqslant n)$	En-route nodes from CH to BS.
t	Maximum number of detectable compromised nodes in the original IHHAS.
v_i $(1 \leqslant i \leqslant \tau)$	Nodes (including CH) in the cluster ($\tau \geqslant t+1$).

A. Overview of IHHAS

IHHAS consists of five phases.

1) Node initialization and Deployment: The key server loads each node with a unique ID and necessary keying materials. After deployment, the node establishes a pairwise key with its one-hop neighbors.

2) Association Discovery: This phase is for a node to discover the IDs of its association nodes. The initial path setup consists of two steps - base station hello and cluster acknowledgment. Incremental association discovery is used for path changes from cluster to base station.

3) Report Endorsement: Requires that at least $t+1$ nodes agree on the report for it to be considered a valid report. Every participating node computes two MACs (Message Authentication Codes) over the event, one using its shared key with BS (called *individual MAC*) and other using the shared key with its upper associated node (called *pairwise MAC*). Then its sends the MACs to CH. CH collects MACs from all the participating nodes, authenticates them, wraps them into a single report and forwards it to BS. The format of the IHHAS report is as follows (assuming $t = 3$):

$$\begin{aligned} R \ : \ & E, C_i, \{v_1, v_2, v_3, CH\}, XMAC(E), \\ & \{MAC(K_{CHu_4}, E), MAC(K_{v_3u_3}, E), \\ & MAC(K_{v_2u_2}, E), MAC(K_{v_1u_1}, E)\}. \end{aligned} \tag{2}$$

where $MAC(K_{v_iu_i}, E), i = 1, 2, 3, 4$ are the pairwise MACs and XMAC is a compressed MAC computed by CH using individual MACs as given below:

$$\begin{aligned} XMAC(E) \ = \ & MAC(K^a_{v1}, E) \oplus MAC(K^a_{v2}, E) \\ \oplus \ & MAC(K^a_{v3}, E) \oplus MAC(K^a_{v4}, E). \end{aligned} \tag{3}$$

4) En-route filtering: Every en-route node verifies the MAC computed by its lower associated node, and then removes the MAC from the received report. If verification succeeds, it attaches a new MAC based on pairwise key with its upper associated node and forwards it to BS.

5) Base station verification: BS verifies the report after receiving it. If the BS detects that $t+1$ nodes have endorsed the report correctly, it accepts the report; otherwise, it discards the report.

B. Limitations of IHHAS

While IHHAS works well with the system model described in [1], it overlooks the following scenarios.

1) Large cluster size not addressed: Designed with the implicit assumption that there are exactly $t+1$ nodes in a cluster (including CH), IHHAS works well as long as the number of compromised nodes (within cluster or en-route) is no larger than t. With more than $t+1$ nodes in the cluster, association discovery phase of IHHAS works incorrectly since it can not guarantee a unique lower association node to an en-route node. We generalize IHHAS to accommodate more than $t+1$ cluster nodes.

2) ID attack not considered: The format of the IHHAS report is given in Eq. (2). All en-route nodes check only the pairwise MACs and do not verify the IDs of sensor nodes endorsing the reports. Only BS can verify the node IDs and the XMAC. However, this makes the scheme prone to *ID attack*, where adversary can simply modify the node ID list $\{v_1, v_2, v_3, CH\}$ so that BS is not able to verify the XMAC with the modified list and all en-route nodes waste energy forwarding such false data. To overcome this limitation, we make a simple improvement to IHHAS wherein each node also includes its node ID in the MAC contents and the en-route nodes verify node ID in the list with that in the MAC contents.

3) Not suitable for distinct reports from sensors: IHHAS works perfectly when all the sensors agree on an event E, which means that E could be a logical or boolean value so that all sensors agree on exactly the same thing. For example, sensors responding either "Yes" or "No" to a query whether the room temperature is higher than 150^oF, would be such an event. In a scenario where CH needs to do aggregation and all the sensors could report possibly different readings and digests which would generally be the case in practice, the computation of XMAC as in Eq. (3) is not possible. Further, it may incur infeasible communication overheads to forward all individual MACs to BS instead of compressing them.

C. Enhancing IHHAS to Integrate with SSTF

To address above inadequacies, following modifications are done to association discovery phase, final report preparation by CH and enroute filtering phase in IHHAS.

1) Association Discovery: There are total $\tau \geqslant t+1$ nodes (including CH in the cluster. As in IHHAS, BS sends *Hello* message to enable a node discover its upper association node. On receiving a *Hello* message from BS, a node attaches its own ID to the *Hello* message before re-broadcasting it. The maximum number of node IDs that are included in *Hello* message is $t+1$. CH divides the cluster nodes (including itself) into $t+1$ groups, g_i $(1 \leqslant i \leqslant t+1)$ and each group has a minimum of 1 and a maximum of $\psi = \lceil \frac{\tau}{t+1} \rceil$ nodes. When CH receives the *Hello* message containing $t+1$ IDs from its previous hop node, it assigns the $t+1$ IDs to the $t+1$ groups. Thus, all the nodes in a group have a single upper association node. Also, CH keeps a list of the nodes in each group.

Example: Fig. 2 illustrates the association discovery process (BS "*Hello*" and cluster "ACK"). There are total 10 nodes. When CH receives the hello message $(u4, u3, u2, u1)$, it divides all the nodes into 4 groups: $(g1(v5, v2, v1), g2(v6, v4, v3), g3(v9, v8, v7), g4(CH))$. There can be a maximum of 3 nodes in a group. Then it assigns each of $(u4, u3, u2, u1)$ to $(g4, g3, g2, g1)$ respectively. During the cluster ACK process, the acks consists of group ID with the group nodes so that the en-route nodes come to know of its lower associated nodes. For example, when $u1$ receives $(g4(CH), g3(v9, v8, v7), g2(v6, v4, v3), g1(v5, v2, v1))$, it knows that its lower associated nodes

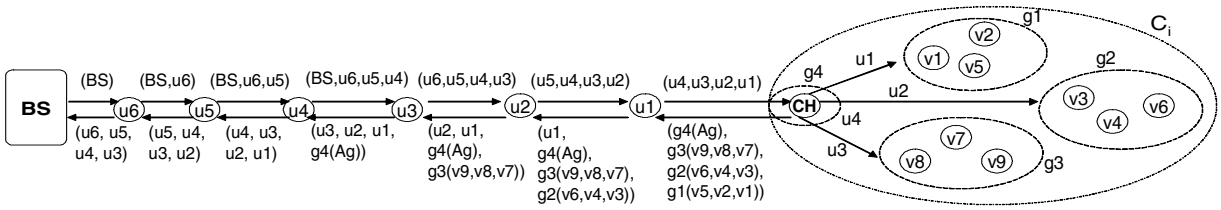


Fig. 2. An example to show the association discovery process for $t+1=4$ and $\tau=10$. The cluster head CH divides the cluster nodes (including itself) into $t+1$ groups and assigns a group ID to each group. Each group has a maximum of $\psi=\lceil\tau/(t+1)\rceil=3$ nodes. BS is the base station. (M) is the content of the beaconing message.

are $v5, v2, v1$ in group $g1$. It then removes this group and substitutes its ID $u1$ in the beginning and forwards $(u1, g4(CH), g3(v9, v8, v7), g2(v6, v4, v3))$ to $u2$. The process proceeds similarly at each en-route node. ■

We define a new term *Report Limit*, θ which is the maximum number of reports from a group that will be used by CH. It is easy to see that $\theta \leqslant \psi$. CH needs at least $t+1$ reports and at most θ reports from each group will be used. To satisfy this, our scheme requires that at least p nodes in the cluster report, where p is given by:

$$p \;=\; max\left(t+1, \psi \times \left(\left\lceil \frac{(t+1)}{\theta} \right\rceil - 1\right) + \theta\right) \tag{4}$$

Note that even though the maximum number of compromised nodes that the scheme can accept is still t; the en-route filtering phase will work only if less than $(t+1)/\theta$ nodes are compromised. If $N_c\left(\frac{t+1}{\theta} \leqslant N_c \leqslant \tau\right)$ nodes are compromised, though BS will eventually detect the false report, the en-route nodes may not be able to detect it and keep on forwarding the false report. θ is useful in the en-route filtering phase, see Section V-C.3.

2) Final Report Preparation: CH sends the aggregated report R_{Ag_i} back to the selected $t+1$ nodes for endorsement. This endorsement prevents CH from lying about the aggregation process. If at sensor node v_k, R_{Ag_i} passes the endorsement tests, it endorses R_{Ag_i} by providing an *individual MAC* for R_{Ag_i} using the authentication key K^a_{vk} shared with BS; and a *pairwise MAC* for (R_{Ag_i}, v_k) using the pairwise key $K_{v_k u_k}$ shared with its upper association node. Note that v_k inserts the node ID into the pairwise MAC contents. Only when CH receives endorsements from all $t+1$ nodes, it generates a compressed MAC over R_{Ag_i} denoted as $XMAC(R_{Ag_i})$ which contains the MACs of the $t+1$ nodes. For example in Fig. 2, if CH receives individual MACs from the previously chosen nodes $v1$, $v2$, $v3$ and $v4$, it computes the XMAC as given below:

$$\begin{aligned} XMAC(R_{Ag_i}) \;=\; & MAC(K^a_{v1}, R_{Ag_i}) \oplus MAC(K^a_{v2}, R_{Ag_i}) \\ & \oplus\; MAC(K^a_{v3}, R_{Ag_i}) \oplus MAC(K^a_{v4}, R_{Ag_i}). \end{aligned} \tag{5}$$

The report R that CH finally generates and forwards to BS consists of the aggregated report R_{Ag_i}, cluster ID C_i, ID list of the $t+1$ endorsing nodes, the XMAC as computed above and the $t+1$ distinct pairwise MACs, and a special counter κ initially set to zero. For example, in Fig. 2, the report R generated by CH is:

$$\begin{aligned} R \;\equiv\; & \langle R_{Ag_i}, C_i, \kappa=0, \{v_1, v_2, v_3, v_4\}, XMAC(R_{Ag_i}), \\ & \{MAC(K_{v_4 u_2}, (R_{Ag_i}, v_4)), MAC(K_{v_3 u_2}, (R_{Ag_i}, v_3)), \\ & MAC(K_{v_2 u_1}, (R_{Ag_i}, v_2)), MAC(K_{v_1 u_1}, (R_{Ag_i}, v_1))\}\rangle. \end{aligned} \tag{6}$$

The order of the pairwise MACs in R corresponds to that in the cluster acknowledgment message during the association discovery phase so that a node receiving R knows which pairwise MACs could be from its lower association nodes. κ is a special counter updated by en-route nodes to keep in track how many consecutive nodes have not been able to verify any of the MACs. It is described more in the En-route Filtering phase next.

3) En-route Filtering: When an en-route node u_k receives R from its downstream node, it checks the number of different pairwise MACs in R. If u_k is s $(s < t+1)$ hops away from BS, it should see at most s pairwise MACs; otherwise, it should see $t+1$ pairwise MACs. u_k tries to verify the last θ MACs in the pairwise MAC list, based on the pairwise key shared with its lower association node(s). The verification is done as follows. u_k checks whether the node ID decrypted from its lower association node MAC is in the ID list of endorsing nodes, and whether report endorsed by its downstream node v_k conforms to R_{Ag_i}. If the node u_k is not able to verify any of the pairwise MACs, it increments κ by 1 and forwards the report R to its upstream node.

If at any point of time $\kappa \geqslant \lceil(t+1)/\theta\rceil$, it implies that more than $\lceil(t+1)/\theta\rceil$ adjacent en-route nodes have been compromised and the report will be dropped. On the other hand, the scheme will not work if more than $\lceil(t+1)/\theta\rceil$ nodes are compromised, since the compromised nodes may reset κ to zero.

If u_k is able to verify $\nu(\leqslant\theta)$ nodes and if u_k is more than $t+1$ hops away from BS, it proceeds to compute ν new pairwise MACs over the report R_{ki} $(1 \leqslant k \leqslant \nu)$ using the pairwise key shared with its upper association node. It then removes the last ν MACs from the MAC

list and inserts the ν new MACs at the beginning of the MAC list. Finally it resets κ to zero and forwards the report to its upstream node.

VI. Security Analysis

Individual sensor nodes or CH can lie about the measurements, digests and aggregated reports. All these attacks are collectively referred to as *content attacks*. In this section, we present a brief analysis on various content attacks. Throughout the analysis, for the sake of simplicity, the index for sliding window in the reports is omitted.

A. Content Attacks by Individual Sensors

1) Effect of False Injection: A pairwise distribution test is performed to test the equality of means reported by the sensor nodes. Let v_j be a compromised node with true mean and variance of (μ, σ^2). Assume v_j reports (μ', σ'^2) instead of the true values. To pass the distribution tests, the following conditions should hold:

$$|\mu' - \mu_k| \leqslant z_\alpha \frac{\sigma'^2 + \sigma_k^2}{\sqrt{w}}; \forall k \in [1, \tau], k \neq j \tag{7}$$

where (μ_k, σ_k^2) is the distribution reported by sensor v_k and τ is the number of nodes in the cluster, w is size of sliding window.

The reading reported by sensor should be within limits to pass the bin test. The compromised node wants a false reading $r' = r + \Delta_r$ to get accepted, where r is the true reading measured by the sensor. We are interested in computing the maximum possible expected distortion that an attacker can inject without being detected i.e. we want to maximize $E[\Delta_r|\Delta_r \text{ is accepted}]$.

We can obtain [13] the optimal $\Delta_r = \Delta_r^*$ as:

$$\Delta_r^* = \begin{cases} 2\sigma'_{Ag} - \mathcal{W}_r, & \mathcal{W}_r < \sigma'_{Ag} \\ \sigma'_{Ag}, & \mathcal{W}_r \geqslant \sigma'_{Ag} \end{cases} \tag{8}$$

where $\mathcal{W}_r = \max(r_i) - \min(r_i)$, r_i being the reported individual sensor readings. σ'^2_{Ag} is the compromised aggregated variance. Also, the maximum expectation of $E[\Delta_r|\Delta_r \text{ is accepted}]$ is given by [13]:

$$E_{max} = \begin{cases} 2\sigma'_{Ag} - \mathcal{W}_r, & \mathcal{W}_r < \sigma'_{Ag} \\ \frac{\sigma'^2_{Ag}}{\mathcal{W}_r}, & \mathcal{W}_r \geqslant \sigma'_{Ag} \end{cases} \tag{9}$$

Fig. 3 illustrates the variation of expectation with respect to Δ_r. We can see that Δ_r is dependent on the aggregated variance σ'^2_{Ag} and $\mathcal{W}_r$. When source variation is less, $\mathcal{W}_r$ is small and compromised node should report $r' = r + 2\sigma'_{Ag} - \mathcal{W}_r$; and in case of a highly varying source, $\mathcal{W}_r$ is large and the compromised node should report $r' = r + \sigma'_{Ag}$. If there are $\mathcal{K}$ nodes participating in the aggregation process, on an average, a single compromised node is able to distort the aggregated reading r_{Ag} by $E_{max}/\mathcal{K}$, where E_{max} is given by Eq. (9). Note that $\mathcal{K} \leqslant \mathcal{E}$, where $\mathcal{E}$ denotes the number of eligible sensors.

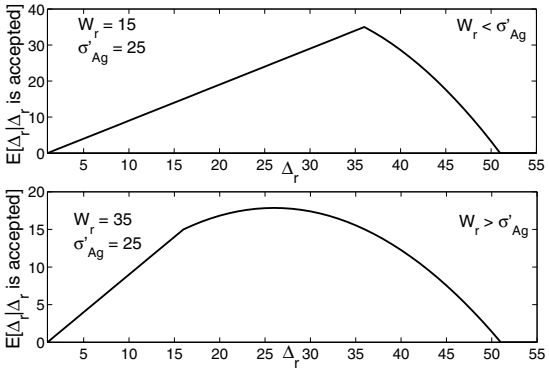


Fig. 3. $E[\Delta_r|\Delta_r \text{ is accepted}]$ vs. Δ_r.

Let r_{Ag} be the genuine aggregated reading. When the adversary injects a false reading $r' = r + \Delta_r$, the aggregated reading r_{Ag} is given by:

$$r_{Ag} = \frac{\sum r_i + r + \Delta_r}{\text{Number of Samples in the Largest Bin}}, \tag{10}$$

where r_i are the genuine readings reported by other sensors. Thus, the impact of the false injection on r_{Ag}, denoted by $\mathcal{F}$, is:

$$\mathcal{F} = \frac{\Delta_r}{\text{Number of Samples in the Largest Bin}}. \tag{11}$$

2) Attack Strategies: There are two strategies that the adversary can adopt to inject false data and distort r_{Ag}. As can be seen from Eq. (11), the adversary can either attempt to maximize Δ_r or minimize the number of samples in the largest bin to increase $\mathcal{F}$.

a) Strategy 1: First strategy is to report a small false variance such that the aggregated variance and hence the bin width is reduced. This is equivalent to decreasing the denominator in Eq. (11). As a result some of the genuine readings are excluded from aggregation and hence, the false data injected by the adversary can have more impact. However, due to reduced bin width, the distortion Δ_r that can be introduced into the reading is also small.

b) Strategy 2: The other strategy is to report a large variance such that the aggregated variance is increased. This results in a larger bin width and hence, larger distortion Δ_r can be introduced in the reading.

3) Selecting the Best Strategy: Let σ_0^* be the genuine aggregated standard deviation and σ_1^* and σ_2^* be the false aggregated standard deviation computed using strategy 1 and strategy 2 respectively.

Consider Fig. 4. Let P_0, P_1, P_2 denote the probability that a sample lies in the bins $A_iD_i(\mu \pm 2\sigma_i^*; i = 0, 1, 2)$

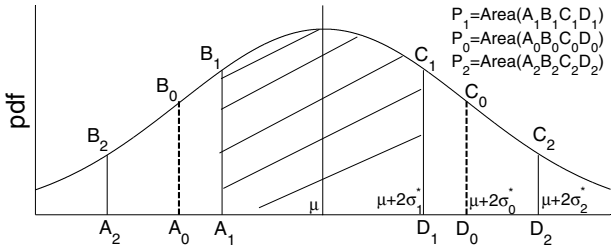


Fig. 4. Demonstrating the relation between bin size and aggregated standard deviation.

for the case of no compromised node, one compromised node using strategy 1 and one compromised node using strategy 2 respectively. $\mathcal{E}$ denotes the number of eligible sensors. The bin size i.e. the expected number of samples that lie in the bins A_0B_0, A_1B_1, A_2B_2 is given by $\mathcal{E}P_0$, $\mathcal{E}P_1$, $\mathcal{E}P_2$ respectively.

From Fig. 4, we can see that:

$$\begin{aligned} \frac{P_1}{P_2} &= \frac{Area(A_1B_1C_1D_1)}{Area(A_2B_2C_2D_2)} > \frac{\sigma_1^*}{\sigma_2^*} \\ &\Rightarrow \frac{\sigma_1^*}{P_1} < \frac{\sigma_2^*}{P_2}. \end{aligned} \tag{12}$$

The adversary introduces a distortion of Δ_r^* according to Eq. (8). For the case when $\Delta_r^* = \sigma_{Ag}^*$, from Eq. (11), the impact on r_{Ag} using the first and second strategies is: $\mathcal{F}_1 = \frac{\sigma_1^*}{\mathcal{E}P_1}$ and $\mathcal{F}_2 = \frac{\sigma_2^*}{\mathcal{E}P_2}$ respectively. From Eq. (12), it can be easily seen that:

$$\frac{\sigma_1^*}{\mathcal{E}P_1} < \frac{\sigma_2^*}{\mathcal{E}P_2}, \tag{13}$$

implying $\mathcal{F}_2 > \mathcal{F}_1$. Hence, the attacker can cause maximum distortion in r_{Ag} when it adopts the second strategy. So, the compromised node should report a high variance. For maximum impact, the adversary reports a fake variance equal to ∞. Similar analysis can be done for the case, when $\Delta_r^* = 2\sigma_{Ag}^* - \mathcal{W}_r$ (refer to Eq. (8)) and it is seen here too that attacker can cause maximum distortion in r_{Ag} by faking its variance as ∞.

Hence, the adversary reports $\sigma^{*2} = \infty$. Consequently, regardless of the μ' being reported, the effect on μ_{Ag}^* and σ_{Ag}^{*2} is given by:

$$\begin{cases} \mu_{Ag}^* = \frac{\sum_{\substack{k=1 \\ k\neq j}}^{\gamma} \mu_k/\sigma_k^2}{\sum_{\substack{k=1 \\ k\neq j}}^{n} 1/\sigma_k^2}, \\ \sigma_{Ag}^{*2} = \left(\sum_{\substack{k=1 \\ k\neq j}}^{\gamma} 1/\sigma_k^2\right)^{-1}, \end{cases} \tag{14}$$

where j is the index of the compromised sensor node.

Further, from Eq. (11), the effect on r_{Ag} is:

$$\mathcal{F} = \frac{\Delta_r^*}{\mathcal{K}}, \tag{15}$$

where $\mathcal{K} = \mathcal{E}P_1$ and Δ_r^* is given by:

$$\Delta_r^* = \begin{cases} 2\sigma_{Ag}^* - \mathcal{W}_r, & \mathcal{W}_r < \sigma_{Ag}^* \\ \sigma_{Ag}^*, & \mathcal{W}_r \geqslant \sigma_{Ag}^* \end{cases} \tag{16}$$

and σ_{Ag}^* is given by Eq. (14).

B. Content Attacks by the Cluster Head

CH produces an aggregated report R_{Ag_i} and sends it back to individual sensors for endorsement. The worst case performance of the system occurs when CH is compromised. This happens because compromised CH can lie about the aggregated report $R_{Ag} = \langle r_{Ag}, \mu_{Ag}, \sigma_{Ag}^2 \rangle$. CH sends R_{Ag} back to selected sensors for endorsement. The following conditions should hold for the R_{Ag} to be accepted for endorsement:

- From Eq. (1), we can see that $\sigma_{Ag} \leqslant min(\sigma_i)$, for each sensor v_i. Hence R_{Ag} with a larger σ_{Ag} will be rejected. To alter μ_{Ag} and r_{Ag}, CH chooses the largest possible σ_{Ag} given by: $\sigma'_{Ag} = min(\sigma_i)$.
- A sensor v_i performs distribution test to test the equality of μ_{Ag} and μ_i. The maximum false μ'_{Ag} that would satisfy the distribution test is given by:

$$\mu'_{Ag} = min\left(\mu_i + z_\alpha \frac{min(\sigma_i^2) + \sigma_i^2}{\sqrt{w}}\right) \tag{17}$$

  where i is the index of the eligible sensors.
- Further, the aggregated reading r_{Ag} should satisfy the bin test at each endorsing sensor v_i. Let r_{Ag} be the true aggregated reading, and r'_{Ag} be the maximum acceptable false reading reported by compromised CH. It is easy to see that, if CH reports $r'_{Ag} = min(r_i) + 2min(\sigma_i)$, it will always be accepted. Thus, CH can distort the true readings by a maximum of $r_{Ag} - min(r_i) + 2min(\sigma_i)$.

Since our security framework is based on IHHAS, our scheme is equally resilient as IHHAS to other security attacks, such as outsider attacks, replay attacks, cluster insider attacks and en-route insider attacks. Due to space limitations, discussions on those security attacks are omitted.

VII. Performance Evaluation

We study the performance of our scheme by simulation. We compare our scheme with a simple majority voting scheme to show the effectiveness of our scheme in preserving transient data. We also demonstrate the limited impact of false data in the presence of compromised nodes under various attack strategies.

A. Simulation Setup

The wireless sensor network is divided into circular clusters. Each cluster is responsible for sensing the time-varying phenomenon in its region. We focus on one particular cluster shown in Fig. 5 to demonstrate our scheme. Cluster nodes are randomly placed in the circular region and one of the nodes is CH. A single source

is present at a random location in the cluster. The phenomenon exhibits a radial diffusion pattern, implying that the sensors nearest to the source sense the change first. Table III lists the parameters used for simulation.

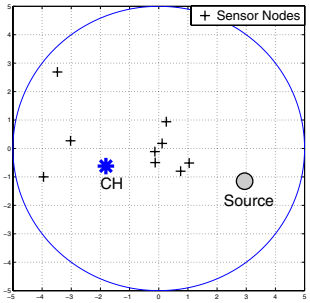


Fig. 5. Simulation Setup.

TABLE III
SIMULATION PARAMETERS

Parameter	Notation	Value
Phenomenon variation rate	ρ	10 units/sec
Maximum inter-sensor distance	d	10 meters
Diffusion rate	$\mathcal{D}$	2 units/sec
Sampling rate	x	10 samples/sec
Reporting interval	n	25 samples
Sliding window size	w	1000 samples
Number of nodes in the cluster	τ	10
Random measurement error at sensors		$\mathcal{N}(0, 0.01)$

B. Simulation Results

We conduct various simulations to demonstrate the effectiveness of SSTF in meeting its design goals viz. preservation of data transience and limiting the impact of false data injection.

1) Preservation of Data Transience: We consider the performance of our scheme in the presence of no compromised nodes. The phenomenon varies from 0 units/second to 50 units/second which amounts to a change of 0 units/sample to 5 units/sample. Fig. 6 illustrates the simulation results. In our scheme most of the times all the genuine data is preserved regardless of transient variations. It is observed that when the variation rate is small, up to 30% of the nodes are excluded from participating in the aggregation. This happens because the bin size becomes very small when the source is constant. However, this doesn't hamper the ability of our scheme to monitor transient data since some genuine nodes are excluded from the largest bin only when source data is itself constant and there is negligible impact on r_{Ag}.

We compare our scheme to a simple majority voting scheme where the nodes agree if the readings reported are within random measurement error of each other. When

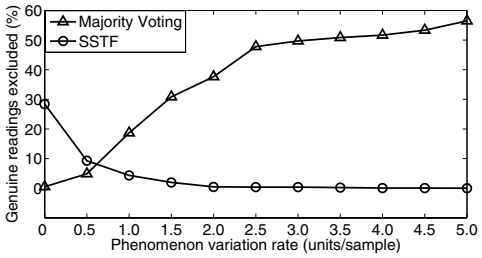


Fig. 6. Percentage of genuine readings excluded vs. Phenomenon variation rate.

there is no variation, the readings are pretty constant and all the nodes agree, however, as the variation rate increases, the readings amongst sensors do not agree with each other anymore, and more and more genuine data are excluded from aggregation. Thus the system starts losing "important" information during data transience which is not desired. We can see in Fig. 6, almost 60% genuine data is lost at high variation rate.

2) Limiting the Impact of False Data Injection: In Fig. 7, we show the effect of false data with respect to different phenomenon variation rates and false injection. X-axis represents the standard deviation of the varying source data which is indicative of phenomenon variation. Y-axis represents the false injection. Z-axis represents the effect on r_{Ag} expressed as a percentage of the source data standard deviation. As discussed above, it can be seen that, for a constant false injection, the impact of false data increases with the phenomenon variation rate. On the other hand for a constant rate, as the false injection is increased, the impact of false data first increases and then decreases and becomes zero as the false injection is increased further. This is attributed to the fact that the false reading remains no longer a part of the largest bin and is excluded from aggregation. It is also observed

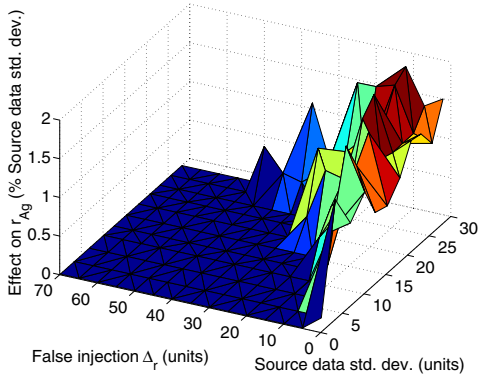
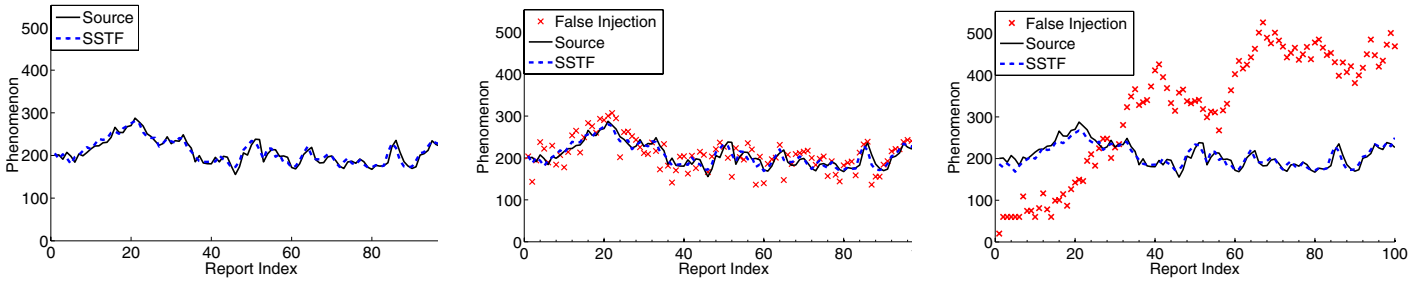


Fig. 7. False effect vs. Phenomenon variation and False injection. Phenomenon variation is indicated by source data standard deviation and z-axis shows the impact on r_{Ag} as a percentage of source data standard deviation.



(a) No Attack (b) Attack satisfying Bin Test (c) Arbitrary Attack

Fig. 8. Demonstrating the effectiveness of SSTF. Shown are two attack strategies. In (b), the attacker chooses to inject false data that complies to the bin test and in (c), the attacker injects arbitrarily random false data.

that impact on r_{Ag} is very limited (up to 2% of source data standard deviation) which conforms to our security analysis (refer to Section VI-A).

Fig. 8 further demonstrates the effectiveness of SSTF in limiting the impact of false data. Shown is a snapshot of a randomly varying source, the Y-axis represents the value in units of the phenomenon. The X-axis represents the report index, a stretch of 100 reports is shown and we can see that SSTF closely resembles the source data. In Fig. 8(a), there is no attack and SSTF is close to the source data, it deviates only due to the delay incurred for the phenomenon to propagate from source to sensor nodes. Fig. 8(b) shows the case when attacker does a subtle attack by reporting false values that pass the distribution test and bin test. However, the impact on the aggregated reading is almost negligible. In Fig. 8(c), the attacker injects arbitrarily random false data, which doesn't alter the result of SSTF either since the injected false data is excluded during the testing process.

VIII. Conclusions and Future Work

We present SSTF, a secure statistical scheme to distinguish data transience from false injection in clustered sensor networks. We develop a statistical framework which is the kernel of SSTF and enhance the IHHAS scheme to be used as the underlying security framework. In contrast to existing false data rejection schemes, SSTF requires each individual sensor to report a statistical digest, in addition to the sensed reading and we emphasize the merits of this strategy to effectively monitor transient variations in the phenomenon. Through detailed security analysis and intensive simulations, we demonstrate the effectiveness of our scheme in preserving the transient data while being resilient to false data injection attacks.

SSTF is designed primarily for applications requiring periodic reporting and monitoring. Future work includes applying SSTF to query-based setup, wherein the sensors respond to a query from the BS, and integration of SSTF with other frameworks such as [4] for operating in a dynamically changing topology or [14] to make it work in a structure-free aggregation setup. We also plan to implement SSTF on a sensor network test-bed.

References

[1] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering false data injection in sensor networks," in *Proc. IEEE SSP*, May 2004.

[2] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2004.

[3] H. Yang and S. Lu, "Commutative cipher based en-route filtering in wireless sensor networks," in *Proc. IEEE VTC*, Sept. 2004.

[4] Z. Yu and Y. Guan, "A dynamic en-route scheme for filtering false data injection in wireless sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2006.

[5] B. Przydatek, D. Song, and A. Perrig, "SIA: Secure information aggregation in sensor networks," in *Proc. ACM SenSys*, Nov. 2003.

[6] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. ACM CCS*, 2002.

[7] Y. Zhang, J. Yang, and H. Vu, "Interleaved authentication for filtering false reports in multipath routing based sensor networks," in *Proc. IEEE IPDPS*, Apr. 2006.

[8] A. Mahimkar and T. Rappaport, "SecureDAV: A secure data aggregation and verification protocol for sensor networks," in *Proc. IEEE GLOBECOM*, Nov. 2004.

[9] D. Wagner, "Resilient aggregation in sensor networks," in *Proc. 2nd ACM SASN*, Oct. 2004.

[10] L. Hu and D. Evans, "Secure aggregation for wireless networks," in *Proc. Workshop on Security and Assurance in Adhoc Networks*, Jan. 2003.

[11] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: A secure hop-by-hop data aggregation protocol for sensor networks," in *Proc. ACM MOBIHOC*, May 2006.

[12] R. Hogg, *Probability and statistical inference*, 2nd ed. Macmillan Publishing Co., Inc, 1983.

[13] V. Shukla and D. Qiao, "A robust statistical scheme to monitor transient phenomenon in sensor networks," To appear in *Proc. ICC*, June 2007.

[14] K. Fan, S. Liu, and P. Sinha, "On the potential of structure-free data aggregation in sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2006.